# Project - Optimal classification trees

Zacharie Ales

2023 - 2024

The objective of this project is to compute optimal classification trees for several datasets using formulation $F$.

## 1 Description of the provided Julia code

The code includes:

- 3 data sets (Iris, Wine and Seeds);
- The definition of formulation $F$ for both axis-aligned and oblique split functions;
- A naive method to compute clusters $\mathcal{C}$ from a dataset;
- Formulations $F_U(\mathcal{C})$, $F_S^e(\mathcal{C})$ and $F_S^h(\mathcal{C})$.

The main files and folders are:

- **File** `main.jl`

  Contains one method `main()` which applies $F$ with both axis-aligned and oblique split functions for several datasets and several tree depths ($D \in \{2, 3, 4\}$).

  To execute the method, simply call `main()` in a Julia console after including the file (`include("main.jl")`).

- **File** `building_tree.jl`

  Contains the definition of the functions required to solve $F$ and $F_U$.

- **File** `main_merge.jl`

  Similar to `main.jl` but applies $F_U$ instead of $F$ for partitions $\mathcal{C}$ of different size obtained through a naive clustering method.
  To execute the method, simply type `main_merge()` in a Julia console after including the file (`include("main_merge.jl")`).

- **File** `main_iterative_algorithm.jl`

  Similar to `main.jl` but applies $F_S^h$ and $F_S^e$ instead of $F$ for the same partitions.
  To execute the method, simply type `main_iterative()` in a Julia console after including the file (`include("main_iterative_algorithm.jl")`).

- **File** `merge.jl`

  Contains the functions:

  - `simpleMerge()`: cluster the data using a naive algorithm described in the course;
  - `exactMerge()`: cluster the data so that hypothesis $H_1$ is satisfied.

    (`exactMerge` is not used in `main_merge()` since it does not regroup any data for the three provided datasets).

- **Folder `data`**

  Contains the datasets.

- **Folder `struct`**

  Contains the structures used in the code (`Tree`, `Cluster` et `Distance`).

## 2 Requested work

Apply methods `main()`, `main_merge()`, and `main_iterative()` on the 3 datasets provided and 2 additional datasets of your choice. For simplicity, only apply `main_iterative()` in the univariate case.

Address at least one of the following open questions:

1. Design and test better clustering methods which must satisfy one of the following conditions:
   - Enable to obtain clusters whose data may be of different classes; or
   - Do not simply consider independently each class. For example, by applying a $k$-means for all the data of each class without taking into account the data of the other classes.
2. Find one or several clustering hypothesis similar to $H_1$ which enable to obtain optimal or near-optimal solutions;
3. Identify and use valid inequalities to improve the solution of formulation $F$ (they can be added initially or dynamically through a callback);
4. Remove constraints from formulation $F$ and add them dynamically during the solution through a callback;
5. Try new shifting algorithms to select the split functions at each step of the iterative algorithm. Evaluate if your method enables to reduce the computation time, the number of iterations or if it enables to obtain better solutions with $F_S^h$.
6. Find and test any ideas which can improve the performances.

## 3 Due date

Send me before the $31^{st}$ of march:

- your files (data, code, ...);
- your report including
  - your results with comments (computation time, accurracy on the different datasets, for different tree depths, for both types of split functions).
  - the description of your work on the open question(s).

The report can be short. You do not have to present the context, Formulation $F$, the different types of separation or the principle of clustering data to find classification trees.